UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/665,564	09/22/2003	Louis R. Degenaro	YOR920030126US1	6151

48150 7590 06/11/2008
MCGINN INTELLECTUAL PROPERTY LAW GROUP, PLLC
8321 OLD COURTHOUSE ROAD
SUITE 200
VIENNA, VA 22182-3817

EXAMINER	
SYED, FARHAN M	

ART UNIT	PAPER NUMBER
2165	

MAIL DATE	DELIVERY MODE
06/11/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/665,564	DEGENARO ET AL.	
	Examiner	Art Unit	
	FARHAN M. SYED	2165	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 September 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-37 are pending.

Reopening Prosecution

In view of the Appeal Brief filed on 24 September 2007, PROSECUTION IS HEREBY REOPENED. New grounds of rejection are set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office Action is non-final) or a reply under 37 CFR 1.113 (if this Office Action is final); or,

(2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid.

A Supervisory Patent Examiner (SPE) has approved of the reopening prosecution by signing below:

Response to Arguments

2. Applicant's arguments with respect to claims 1-37 have been considered but are moot in view of the new ground(s) of rejection.

The arguments are deemed moot since it appears that the rejection previously made has been misinterpreted. The examiner has set forth new grounds of rejection in light of the Appellant’s arguments presented.

The Examiner’s rejections of the claims, now set forth are in light of the applicant’s arguments against the art applied, But applied in the modified position therefore, the arguments are deemed moot.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 14-26 and 32 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5. Regarding claims 14-26, the word "means" is preceded by the word(s) "comprises" in an attempt to use a "means" clause to recite a claim element as a means for performing a specified function. However, since no function is specified by the word(s) preceding "means," it is impossible to determine the equivalents of the element, as required by 35 U.S.C. 112, sixth paragraph. See *Ex parte Klumb*, 159 USPQ 694 (Bd. App. 1967).

6. Claim 32 provides for the use of a computer-readable medium, but, since the claim does not set forth any steps involved in the method/process, it is unclear what method/process applicant is intending to encompass. A claim is indefinite where it merely recites a use without any active, positive steps delimiting how this use is actually practiced. That is, claim 32 appears to contain a product and process in the same claim, where it is directed towards the use of a computer-readable medium and the method of using it.

Claim 32 is rejected under 35 U.S.C. 101 because the claimed recitation of a use, without setting forth any steps involved in the process, results in an improper definition of a process, i.e., results in a claim which is not a proper process claim under 35 U.S.C. 101. See for example *Ex parte Dunki*, 153 USPQ 678 (Bd.App. 1967) and *Clinical Products, Ltd.* v. *Brenner*, 255 F. Supp. 131, 149 USPQ 475 (D.D.C. 1966).

7. The Examiner notes that claim 27 is directed to a system comprised of plurality of actual resources, but the use of a service within the system. It appears to the Examiner that the Applicant is attempting to describe a product-by-process claim and does not believe claim 27 and the depending claims 28-30 are indefinite under 35 U.S.C. 112, 2nd paragraph. See MPEP 2173.05(p).

Claim Rejections - 35 USC § 101

8. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

9. Claim 32 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claim 32 recites "a computer-readable medium tangibly embodying a program of machine-readable instruction.." The Applicant defines "machine-readable instructions" in the specification, see page 39, lines 16-17, as software object code, compiled from a language such as 'C'., etc." Although the Applicant does not specifically define computer-readable medium, the Examiner interprets computer-readable medium to be a variety of machine-readable data storage media, which, according to the Applicant's specification, see page 39, lines 9-15, to include "...other suitable signal-bearing media including transmission media such as digital and analog and communication links and wireless...".

The claim lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, functional descriptive material *per se*.

Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*, 33 F.3d at 1360, 31 USPQ2d at 1759. When <u>functional</u> descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive

material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994)

Merely claiming nonfunctional descriptive material, i.e., abstract ideas, stored on a computer-readable medium, in a computer, or on an electromagnetic carrier signal, does not make it statutory. See *Diehr*, 450 U.S. at 185-86, 209 USPQ at 8 (noting that the claims for an algorithm in *Benson* were unpatentable as abstract ideas because "[t]he sole practical application of the algorithm was in connection with the programming of a general purpose computer.").

Claim Rejections - 35 USC § 103

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

11. Claims 1-37 are rejected under 35 U.S.C. 102(a) as being anticipated by a non-patent literature entitled "XTABLES: Bridging relational technology and XML" by J.E. Funderburk, G. Kiernan, J. Shanmugasundaram, E. Shekita, and C. Wei, pages 616-641 (known hereinafter as Funderburk)(previously presented) in view of Loaiza et al., (U.S. Patent 6,618,822 B1 and known hereinafter as Loaiza).

As per claims 1, 14, 31, and 32, Funderburk teaches a method of claim 1 (Abstract), a system of claim 14 (Abstract), a method of claim 31 (Abstract), and a computer-

readable medium of claim 32 (Figure 5) of developing actual resources without alteration into a collection of virtual resources customized to a particular audience, said method comprising (i.e. *"One of the features provided by XTABLES is the ability to create XML views of existing relational data." "Users can then create application-specific XML views on top of the default XML view."* The preceding text clearly indicates that a collection of virtual resources is a specific XML view and existing relational data is the actual resource.)(Page 616, paragraph 2): constructing at least one virtual resource independent of an actual resource (i.e. *"Figure 2 illustrates the default view for a simple purchase-order database. The database consists of three tables, one to track customer orders, a second to track items associated with an order, and a third to track the payments due for each order. Items and payments are related to orders by an order identifier (oid). In the default XML view, top-level elements correspond to tables, with table names appearing as tags. Row elements are nested under these. Within a row element, column names appear as tags and column values appear as text. Although not shown, an XML schema associated with the default view captures primary- and foreign-key relationships."* The preceding text clearly indicates that at least one virtual resource is the default XML view.)(Page 620); connecting the actual resource to the at least one virtual resource (i.e. *"Figure 2 illustrates the default view for a simple purchase-order database. The database consists of three tables, one to track customer orders, a second to track items associated with an order, and a third to track the payments due for each order. Items and payments are related to orders by an order identifier (oid). In the default XML view, top-level elements correspond to tables, with table names appearing as tags. Row elements are nested under these. Within a row element, column names appear as tags and column values appear as text. Although not shown, an XML schema associated with the default view captures primary- and foreign-key relationships."* The preceding text clearly indicates that connecting is corresponding, at least one actual resource is the database consisting of three tables, and virtual resource is the default XML view.)(Page 620); retrieving the at least one virtual resource (i.e. *"Figure 2 illustrates the default view for a simple purchase-order database. The database consists of*

three tables, one to track customer orders, a second to track items associated with an order, and a third to track the payments due for each order. Items and payments are related to orders by an order identifier (oid). In the default XML view, top-level elements correspond to tables, with table names appearing as tags. Row elements are nested under these. Within a row element, column names appear as tags and column values appear as text. Although not shown, an XML schema associated with the default view captures primary- and foreign-key relationships." The preceding text clearly indicates that retrieving is displaying the default XML view)(Page 620); and extracting at least one descriptor from said at least one retrieved virtual resource (i.e. *"Figure 2 illustrates the default view for a simple purchase-order database. The database consists of three tables, one to track customer orders, a second to track items associated with an order, and a third to track the payments due for each order. Items and payments are related to orders by an order identifier (oid). In the default XML view, top-level elements correspond to tables, with table names appearing as tags. Row elements are nested under these. Within a row element, column names appear as tags and column values appear as text. Although not shown, an XML schema associated with the default view captures primary- and foreign-key relationships."* The preceding text clearly indicates that at least one descriptor is a tag.) (Page 620).

However Funderburk does not explicitly teach virtual resource.

Loaiza teaches virtual resource (i.e. *"Alternatively, user-defined functions are registered with the database system to create* ***"virtual tables" that create a view of data in the recovery logs****. The user-defined functions dynamically retrieve and populate column values for a virtual table from underlying data sources."* In the Applicant's specification, see page 3, lines 1-3, where the applicant defines a resource as *"**resource might be a database table**, a Java.RTM. Bean, an Enterprise Java.RTM. Bean (EJB), a Java.RTM. object, a legacy application, a Web Service, a flat file, an eXtensible Markup Language (XML) file, etc."* Therefore, the Examiner interprets virtual tables as virtual resource.)(Column 5, lines 37-42).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Funderburk with the teachings of Loaiza to include virtual resource with the motivation to allow users to query seamlessly over relational data and meta-data and allow users to write queries that span XML documents and XML views of relational data (Funderburk, Abstract).

As per claims 2 and 15, Funderburk teaches a method of claim 2 (Abstract), a system of claim 15 (Abstract) wherein said connecting comprises directly mapping the at least one actual resource to the at least one virtual resource (i.e. *"XTABLES does this by automatically mapping the schema and data of the underlying relational database system to a low-level default XML view."* The preceding text clearly indicates that at least one actual resource is the underlying relational database system.)(Page 616).

However Funderburk does not explicitly teach virtual resource.

Loaiza teaches virtual resource (i.e. *"Alternatively, user-defined functions are registered with the database system to create **"virtual tables" that create a view of data in the recovery logs**. The user-defined functions dynamically retrieve and populate column values for a virtual table from underlying data sources."* In the Applicant's specification, see page 3, lines 1-3, the applicant defines a resource as *"**resource might be a database table**, a Java.RTM. Bean, an Enterprise Java.RTM. Bean (EJB), a Java.RTM. object, a legacy application, a Web Service, a flat file, an eXtensible Markup Language (XML) file, etc."* Therefore, the Examiner interprets virtual tables as virtual resource.)(Column 5, lines 37-42).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Funderburk with the teachings of Loaiza to include virtual resource with the motivation to allow users to query seamlessly over

relational data and meta-data and allow users to write queries that span XML documents and XML views of relational data (Funderburk, Abstract).

As per claims 3 and 16, Funderburk teaches a method of claim 3 (Abstract), a system of claim 16 (Abstract) wherein the constructing comprises at least one of: renaming a method; hiding a method; composing a method; renaming an attribute; hiding an attribute; composing an attribute; assigning to at least one domain; designating as a collection; assigning to at least one validator; assigning a description; designating as at least one of ready and not ready; and assigning a last modified date and time (i.e. *"A query is initially parsed and converted from XQuery to an intermediate query representation called the XML Query Graph Model (XQGM). The query is then composed with the XML views it references, and rewrite optimizations are performed to eliminate the construction of intermediate XML fragments, unroll recursion, and push down predicates."* The preceding text clearly indicates that renaming a method, hiding a method, etc. is example of XML fragments)(Page 622).

As per claims 4, 8, 17 and 21, Funderburk teaches a method of claim 4 and 8 (Abstract), a system of claim 17 and 21 (Abstract), wherein said at least one virtual resource comprises a plurality of virtual resources and said virtual resources are connected to each other through a relationship carrying semantic that can be leveraged by a consumer of resources, said method further comprising (i.e. *"Users can then create application-specific XML views on top of the default XML view. These application-specific views are created using XQuery, a general-purpose, declarative XML query language currently being standardized by the W3C (World Wide Web Consortium)."* The preceding text clearly indicates that a relationship carrying semantic is the creation of application-specific XML views on top of the default XML view, where

XML views are virtual resources and consumer of resources are users.)(Page 616): constructing at least one virtual relationship between at least two virtual resources (i.e. *"Continuing the example, suppose that a user wants to publish the purchase-order database as a list of orders in the XML format shown in Figure 3. There, each order appears as a top-level element, with its associated items and payments (ordered by due date) nested under it. To transform the default view into the desired XML format, a user-defined view called "orders" is created, as shown in Figure 4. The view definition is fairly straightforward. An XQuery FLWR (for, let, where, return) expression (lines 2–22) is used to construct each order element. The "for" clause on line 2 causes the variable $order to be bound to each "row" element of the order table. The XPath expression appearing in line 2 describes how to extract each "row" element from the order table: start at the root of the default view, navigate to each "order" element nested under it, and then navigate to each "row" element nested under those "order" elements. The constructor for each new "order" element is given in lines 4–22. For a given order, nested FLWR expressions are used to construct its list of associated items (lines 6–13) and payments (lines 14–21). The predicate on line 8 ($order/id_$item/oid) is used to join an order with its items. Similarly, the predicate on line 16 ($order/id _ $payment/oid) is used to join an order with its payments."* The preceding text clearly indicates that at least one virtual relationship is orders and the two virtual resources are the default XML view and the user-defined view.)(Page 620-621); coupling at least one actual relationship implementor to at least one virtual relationship (i.e. Per the preceding text, the orders, which is a virtual relationship is based on one of the three tables, where an actual relationship is the customer order.)(Page 620); performing at least one retrieval of a virtual relationship (i.e. *"An XQuery FLWR (for, let, where, return) expression (lines 2–22) is used to construct each order element."* The preceding text illustrates a retrieval of orders from the purchase-order database to the user-defined XML view 'Order.')(Page 620); and extracting at least one descriptor from at least one retrieved virtual relationship (i.e. Per the preceding text, the XML tags, which are descriptors, are embedded in an XML view.)(Page 620).

However Funderburk does not explicitly teach virtual resource.

Loaiza teaches virtual resource (i.e. *"Alternatively, user-defined functions are registered with the database system to create **"virtual tables" that create a view of data in the recovery logs**. The user-defined functions dynamically retrieve and populate column values for a virtual table from underlying data sources."* In the Applicant's specification, see page 3, lines 1-3, where the applicant defines a resource as *"**resource might be a database table**, a Java.RTM. Bean, an Enterprise Java.RTM. Bean (EJB), a Java.RTM. object, a legacy application, a Web Service, a flat file, an eXtensible Markup Language (XML) file, etc."* Therefore, the Examiner interprets virtual tables as virtual resource.)(Column 5, lines 37-42).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Funderburk with the teachings of Loaiza to include virtual resource with the motivation to allow users to query seamlessly over relational data and meta-data and allow users to write queries that span XML documents and XML views of relational data (Funderburk, Abstract).

As per claims 5, 9, 18, and 22, Funderburk teaches a method of claim 5 and 9 (Abstract), a system of claim 18 and 22 (Abstract) wherein said coupling comprises: directly mapping said at least one actual relationship implementor to said at least one virtual relationship (i.e. *"As shown, each edge is uniquely identified by the identifier fields of the source and destination nodes (the sid and did fields). Each edge also contains the name, value, and type information about its destination node. The order among sibling subelements is captured using the ordinal field. In our example, the edge pointing to the root XML element ("PurchaseOrder") is mapped to the first row. Its sid field is 0, which represents the identifier of the document root. The edges pointing to the BuyerName and Date attributes of the "PurchaseOrder" element are mapped to the second and third row, respectively.*

Note that these are related to the purchase order using the sid field. Similarly, the "ItemsBought" and "Payments" subelements of a "PurchaseOrder" element are represented by the fourth and fifth row, respectively. The ordinal field captures their relative order. The other edges of the document are stored similarly." The preceding text clearly indicates that the virtual relationship is the purchase order and the actual relationship implementor is the first row of a table.)(Page 636).

As per claims 6, 10 19, and 23, Funderburk teaches a method of claim 6 and 10 (Abstract), a system of claim 19 and 23 (Abstract) wherein the relationship constructing comprises at least one of: assigning a root virtual resource name; assigning a target virtual resource name; assigning a relationship name; assigning a relationship type; assigning a description; assigning a target instance naming scheme; designating as at least one of ready and not ready; and assigning a last modified date and time (i.e. *"As shown, each edge is uniquely identified by the identifier fields of the source and destination nodes (the sid and did fields). Each edge also contains the name, value, and type information about its destination node. The order among sibling subelements is captured using the ordinal field. In our example, the edge pointing to the root XML element ("PurchaseOrder") is mapped to the first row. Its sid field is 0, which represents the identifier of the document root. The edges pointing to the BuyerName and Date attributes of the "PurchaseOrder" element are mapped to the second and third row, respectively. Note that these are related to the purchase order using the sid field. Similarly, the "ItemsBought" and "Payments" subelements of a "PurchaseOrder" element are represented by the fourth and fifth row, respectively. The ordinal field captures their relative order. The other edges of the document are stored similarly."* The preceding text clearly indicates that the 'PurchaseOrder' is assigning a relationship name between a virtual resource, which is the XML view, and the actual resource, which is a table within a relational database.)(Page 636).

As per claims 7, 12, 20 and 25, Funderburk teaches a method of claim 7 and 12 (Abstract), a system of claim 20 and 25 (Abstract) wherein the retrieving comprises locating virtual relationships by at least one of: a domain; a name; a root; a type; and a target (i.e. *"As shown, each edge is uniquely identified by the identifier fields of the source and destination nodes (the sid and did fields). Each edge also contains the name, value, and type information about its destination node. The order among sibling subelements is captured using the ordinal field. In our example, the edge pointing to the root XML element ("PurchaseOrder") is mapped to the first row. Its sid field is 0, which represents the identifier of the document root. The edges pointing to the BuyerName and Date attributes of the "PurchaseOrder" element are mapped to the second and third row, respectively. Note that these are related to the purchase order using the sid field. Similarly, the "ItemsBought" and "Payments" subelements of a "PurchaseOrder" element are represented by the fourth and fifth row, respectively. The ordinal field captures their relative order. The other edges of the document are stored similarly."* The preceding text clearly indicates that the 'PurchaseOrder' is an example of a name.)(Page 636).

As per claims 11 and 24, Funderburk teaches a method of claim 11 (Abstract), a system of claim 24 (Abstract) wherein the retrieving comprises locating virtual resources by at least one of (i.e. *"Continuing the example, suppose that a user wants to publish the purchase-order database as a list of orders in the XML format shown in Figure 3." The preceding text clearly indicates that retrieving is publishing.)(Page 620): a domain; a name; and a relationship. (i.e. "To transform the default view into the desired XML format, a user-defined view called "orders" is created, as shown in Figure 4."* The preceding text clearly indicates that a user-defined view, which is an example of a virtual resource, called "orders" is, which is an example of name.)(Page 620).

As per claims 13 and 26, Funderburk teaches a method of claim 13 (Abstract), a system of claim 26 (Abstract), wherein descriptor validator information is employed to limit actual resource usage (i.e. *"Another feature provided by XTABLES is the ability to query XML views of relational data. This is important because users often need only a subset of a view's data. Moreover, users often need to synthesize and extract data from multiple views. In XTABLES, queries are specified using XQuery, the same language used to specify XML views. XTABLES executes queries efficiently by performing XML view composition so that only the desired relational data items are materialized."* The preceding text clearly indicates that a descriptor validator information is an XML tag that is used to generate XML views based on the relational data to synthesize and extract data from multiple views, thus limiting the actual resource, which is the relational data, usage.) (Pages 616-617).

As per claim 27, Funderburk teaches a system comprised of a plurality of actual resources, a service to manage descriptions of said actual resources, said service comprising: defining at least one virtual domain to satisfy a user-requirements analysis (i.e. *"Users can query XML documents using the same query language that they use to create and query XML views of relational data. In addition, users can issue queries that span XML documents and XML views of relational data. As a result, users are provided with unified access to both relational data and XML documents, without having to deal with separate databases."* The preceding text clearly indicates that a requirement analysis is a query and a virtual domain is an XML document.)(Page 617); and defining at least one virtual resource describing as least one actual resource within the at least one virtual domain to satisfy the user-requirements analysis (i.e. *"Continuing the example, suppose that a user wants to publish the purchase-order database as a list of orders in the XML format shown in Figure 3. There, each order appears as a top-level element, with its associated items and payments (ordered by due date) nested under it. To transform the default view into the desired XML format, a user-defined view called "orders" is created, as shown in Figure 4."* The preceding text clearly

indicates that at least one virtual resource is the XML view of orders and one actual resource is the purchase-order database.)(Page 620).

However Funderburk does not explicitly teach virtual resource.

Loaiza teaches virtual resource (i.e. *"Alternatively, user-defined functions are registered with the database system to create **"virtual tables" that create a view of data in the recovery logs**. The user-defined functions dynamically retrieve and populate column values for a virtual table from underlying data sources."* In the Applicant's specification, see page 3, lines 1-3, where the applicant defines a resource as *"**resource might be a database table**, a Java.RTM. Bean, an Enterprise Java.RTM. Bean (EJB), a Java.RTM. object, a legacy application, a Web Service, a flat file, an eXtensible Markup Language (XML) file, etc."* Therefore, the Examiner interprets virtual tables as virtual resource.)(Column 5, lines 37-42).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Funderburk with the teachings of Loaiza to include virtual resource with the motivation to allow users to query seamlessly over relational data and meta-data and allow users to write queries that span XML documents and XML views of relational data (Funderburk, Abstract).

As per claim 28, Funderburk teaches a system further comprising: analyzing a requirement for actual resource usage, to provide said user requirements analysis (i.e. *"The XTABLES default XML view captures both relational data and meta-data (schema) information. This allows users to write queries (and create views) that treat relational data and meta-data interchangeably."* The preceding text clearly indicates that the actual resource usage is the relational data and the requirement analysis is the query.) (Page 618).

As per claim 29, Funderburk teaches a system further comprising: defining at least one virtual relationship between at least two virtual resources (i.e. *"To transform the default view into the desired XML format, a user-defined view called "orders" is created, as shown in Figure 4. "* The preceding text clearly indicates that at least one virtual relationship, which is 'orders' exists between two virtual resources, which are the default XML view and the "orders" XML view.)(Page 620).

However Funderburk does not explicitly teach virtual resource.

Loaiza teaches virtual resource (i.e. *"Alternatively, user-defined functions are registered with the database system to create **"virtual tables" that create a view of data in the recovery logs**. The user-defined functions dynamically retrieve and populate column values for a virtual table from underlying data sources."* In the Applicant's specification, see page 3, lines 1-3, where the applicant defines a resource as *"**resource might be a database table**, a Java.RTM. Bean, an Enterprise Java.RTM. Bean (EJB), a Java.RTM. object, a legacy application, a Web Service, a flat file, an eXtensible Markup Language (XML) file, etc."* Therefore, the Examiner interprets virtual tables as virtual resource.)(Column 5, lines 37-42).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Funderburk with the teachings of Loaiza to include virtual resource with the motivation to allow users to query seamlessly over relational data and meta-data and allow users to write queries that span XML documents and XML views of relational data (Funderburk, Abstract).

As per claim 30, Funderburk teaches a system wherein at least one of a virtual resource and a virtual relationship is utilized to create an application program (i.e. *"Once the 'orders' view has been created, queries can be issued against it."* The preceding text clearly indicates

that an application program, which is the ‘orders’ view has been created, since queries can be issued against it, that is a set of instruction codes used to perform some tangible result.)(Page 621).

However Funderburk does not explicitly teach virtual resource.

Loaiza teaches virtual resource (i.e. *“Alternatively, user-defined functions are registered with the database system to create* ***“virtual tables” that create a view of data in the recovery logs****. The user-defined functions dynamically retrieve and populate column values for a virtual table from underlying data sources.”* In the Applicant’s specification, see page 3, lines 1-3, where the applicant defines a resource as *“****resource might be a database table****, a Java.RTM. Bean, an Enterprise Java.RTM. Bean (EJB), a Java.RTM. object, a legacy application, a Web Service, a flat file, an eXtensible Markup Language (XML) file, etc.”* Therefore, the Examiner interprets virtual tables as virtual resource.)(Column 5, lines 37-42).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Funderburk with the teachings of Loaiza to include virtual resource with the motivation to allow users to query seamlessly over relational data and meta-data and allow users to write queries that span XML documents and XML views of relational data (Funderburk, Abstract).

As per claim 33, Funderburk teaches a method of developing actual resources without alteration into a collection of virtual resources customized to a particular audience, said method comprising (i.e. *“As a starting point, XTABLES automatically creates the default XML view, which is a low-level XML view of the underlying relational database ”* The preceding text clearly indicates that refactoring actual resources without alteration into a collection of virtual resource is the creation of a default XML view, where the XML view is the virtual resource and the

database table is the actual resource.)(Page 620): constructing at least one virtual resource independent of an actual resource; and providing in the at least one virtual resource a structured meta-data layer which contains semantic information for leveraging by a consumer of the virtual resources (i.e. *"Users can then create application-specific XML views on top of the default XML view."* The preceding text clearly indicates that a structured a consumer, which is the user, can leverage, which is create application-specific XML views, from the virtual resource, which is the default XML view.)(Page 616).

However Funderburk does not explicitly teach virtual resource.

Loaiza teaches virtual resource (i.e. *"Alternatively, user-defined functions are registered with the database system to create* ***"virtual tables" that create a view of data in the recovery logs****. The user-defined functions dynamically retrieve and populate column values for a virtual table from underlying data sources."* In the Applicant's specification, see page 3, lines 1-3, where the applicant defines a resource as *"****resource might be a database table****, a Java.RTM. Bean, an Enterprise Java.RTM. Bean (EJB), a Java.RTM. object, a legacy application, a Web Service, a flat file, an eXtensible Markup Language (XML) file, etc."* Therefore, the Examiner interprets virtual tables as virtual resource.)(Column 5, lines 37-42).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Funderburk with the teachings of Loaiza to include virtual resource with the motivation to allow users to query seamlessly over relational data and meta-data and allow users to write queries that span XML documents and XML views of relational data (Funderburk, Abstract).

As per claim 34, Funderburk teaches a method wherein said semantic information includes relationships with agreed upon semantics including any of “related-to,” “contains,” and “is-conflicting-with,” between entities (i.e. *“The table and view operators in XQGM are used to refer to relational tables and XML view definitions, respectively. The unnnest operator is used to unnest XML lists. The function operator us used to invoke Xquery-valued functions represented in XQGM can be found in Reference 7.”*)(Page 623).

As per claim 35, Funderburk teaches a method wherein said semantic information allows any of making new resources manipulation operations available to logic authoring tools and services as an input to a conflict detection tool (i.e. *”XTABLES allows users to treat XML document views like XML views of relational data because, internally, XML documents are nothing by XML views of relational data. Whenever a user creates an XML document view, XTABLES uses one of possibly many relational schema generators to automatically create relational tables for storing XML documents. XML documents ‘stored’ in this view are then shredded and stored as rows in these tables. In addition, XTABLE generates a reconstruction XML view over the created relational tables, which (virtually) reconstructs the ‘stored’ XML documents from the shredded rows. A reconstruction XML view is specified just like any other XML view of relational data – by using a query over the default XML view of the created tables.”*)(Page 631).

As per claim 36 and 37, Funderburk teaches a method further comprising: creating at least one virtual resource instance (i.e. “As a starting point, XTABLES automatically creates the default XML view, which is a low-level XML view of the underlying relational database.” The preceding text clearly indicates that at least one virtual resource instance is created, which is the default XML view.)(Page 620); assigning an identity to the at least one virtual resource instance (i.e.

"To transform the default view into the desired XML format, a user-defined view called 'orders' is created, as shown in Figure 4." The preceding text clearly indicates that an identity, 'orders' is assigned to the virtual resource instance, which is the user-defined XML view.)(Page 620); and associating the at least one virtual resource instance with one virtual resource (i.e. The term 'order' is assigned to the XML view.)(Page 620).

However Funderburk does not explicitly teach virtual resource.

Loaiza teaches virtual resource (i.e. *"Alternatively, user-defined functions are registered with the database system to create* ***"virtual tables" that create a view of data in the recovery logs****. The user-defined functions dynamically retrieve and populate column values for a virtual table from underlying data sources."* In the Applicant's specification, see page 3, lines 1-3, where the applicant defines a resource as *"**resource might be a database table**, a Java.RTM. Bean, an Enterprise Java.RTM. Bean (EJB), a Java.RTM. object, a legacy application, a Web Service, a flat file, an eXtensible Markup Language (XML) file, etc."* Therefore, the Examiner interprets virtual tables as virtual resource.)(Column 5, lines 37-42).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Funderburk with the teachings of Loaiza to include virtual resource with the motivation to allow users to query seamlessly over relational data and meta-data and allow users to write queries that span XML documents and XML views of relational data (Funderburk, Abstract).

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

6,947,950 B2	Murthy, Ravi et al	9-2005

2005/0120160 A1	Plouffe, Jerry et al	6-2005
2005/0044301 A1	Vasilevsky, Alexander et al	2-2005

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Farhan M. Syed whose telephone number is 571-272-7191. The examiner can normally be reached on 8:30AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Christian Chace can be reached on 571-272-4190. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/F. M. S./
Examiner, Art Unit 2165

/Christian P. Chace/

Supervisory Patent Examiner, Art Unit 2165